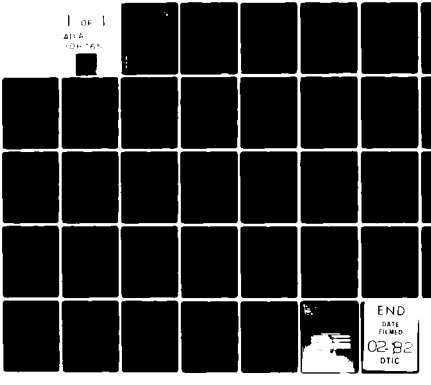
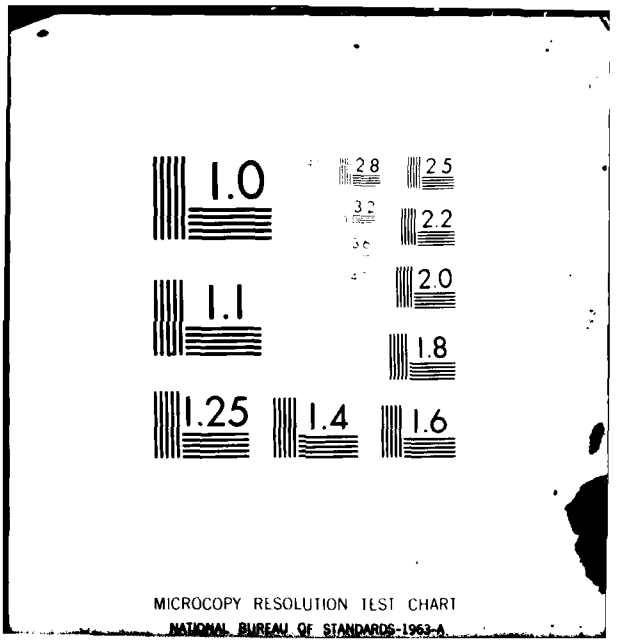
AD-A108 765 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 12/1
ACCELERATION OF CONVERGENCE OF A VECTOR SEQUENCE BY REDUCED RAN--ETC(U)
NOV 81 R P EDDY
UNCLASSIFIED DTNSRDC-81/084 NL




1.0
2.8
2.5
3.2
2.2
3.6
2.0
1.1
1.8
1.25
1.4
1.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

DAVID W. TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER

Bethesda, Maryland 20084

ACCELERATION OF CONVERGENCE OF A VECTOR SEQUENCE BY REDUCED RANK EXTRAPOLATION

by

Robert P. Eddy

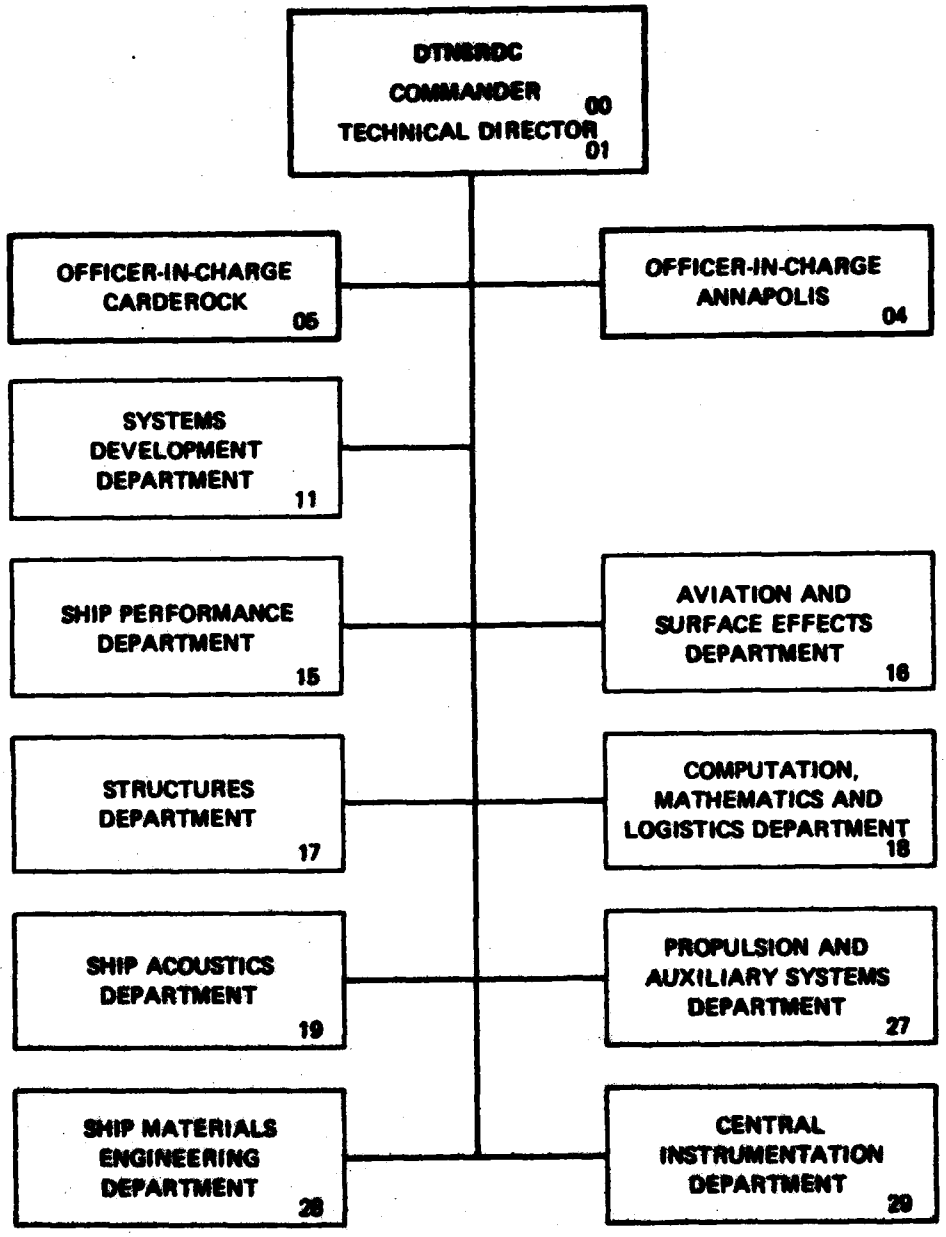
APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

COMPUTATION, MATHEMATICS AND LOGISTICS DEPARTMENT
RESEARCH AND DEVELOPMENT REPORT

November 1981

DTNSRDC-81/084

MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: DTNSRDC-81/084	2. GOVT ACCESSION NO.: AD-A108765
	3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): ACCELERATION OF CONVERGENCE OF A VECTOR SEQUENCE BY REDUCED RANK EXTRAPOLATION	5. TYPE OF REPORT & PERIOD COVERED: Final Report 1975-1980
	6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): Robert P. Eddy	8. CONTRACT OR GRANT NUMBER(s):
9. PERFORMING ORGANIZATION NAME AND ADDRESS: David W. Taylor Naval Ship Research and Development Center Bethesda, Maryland 20084	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: (See reverse side)
11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE: November 1981
	13. NUMBER OF PAGES: 57
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

An early version of this material was presented at the First Annual Workshop on the Information Linkage Between Applied Mathematics and Industry held at the Naval Postgraduate School, Monterey, California, February 23-25, 1978 and appears in the Proceedings thereof, see References.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Acceleration of Convergence
Extrapolation of Vectors
Vector Sequences

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A new family of methods, called reduced rank extrapolation, is developed for accelerating convergence of the sequence of vectors generated during the iterative solution of a system of m linear algebraic equations in m unknowns. Large systems of this kind arise, for example, in the finite difference or finite element solution of partial differential equations.

(Continued on reverse side)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SP.C14.3

(Block 10)

Math Sciences Research Program Program Element 61153N
Task Area SR 0140301 Work Unit 1808-010
(Math Sciences) NAVSEA

(Block 20 continued)

Reduced rank extrapolation is derived from full rank extrapolation, which is a straightforward generalization to vector space of the well known Aitken Δ^2-Shanks e_1, scalar extrapolation. It is applicable when the iteration has reached a point where only a few, say r, eigenvalues dominate the situation and hence only r difference vectors can be linearly independent to specified tolerance. The rank, r, is determined during the solution of an auxiliary problem of best approximation in vector space, i.e., "best" in the sense of minimizing some specified vector norm. Least squares theory, corresponding to the Euclidean norm, is developed in detail herein.

Application to Laplace's equation in a square and in a cube yielded reduction in computation time by a factor ranging between 2.4 and 4.7, and reduction in iteration count by a factor ranging between 3.6 and 5.4.

delta sq. - Shanks e sub 1

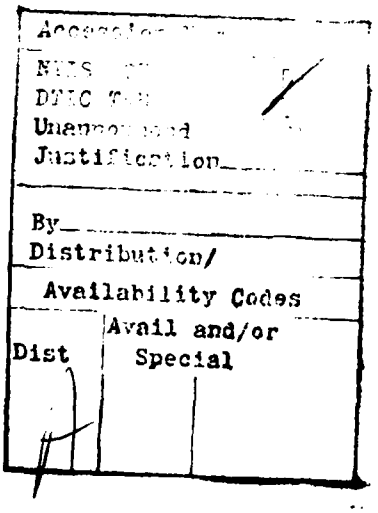


TABLE OF CONTENTS

LIST OF TABLES

NOTE:

This report was prepared for publication after the author's retirement. Accordingly, a final detailed proofreading by the author was not possible.

NOTATION

A	The matrix in the linear vector iteration (Equation (1.1))
b	Constant vector in the linear vector iteration
H	Rectangular matrix, the columns of which are successive second difference vector, h
H^*	Transpose of H
H^+	Generalized inverse h
h	Second difference vector
h_0	Initial second difference vector of a sequence
I	Identity matrix
J	Jordan Canonical form for A
ℓ_1	(Unnamed) norm for vectors (see Equation (4.13))
ℓ_2	Euclidean norm for vectors (see Equation (4.14))
ℓ_∞	Chebyshev norm for vectors (see Equation (4.15))
m	Dimension of vector space = number of equations in iteration system Equation (1.1)
n	Iteration index
R	Residual vector (Equation (4.7))
r	Rank of matrix H and of extrapolation process
T	Triangular square route of H* H (Equation (4.19))
U	Rectangular matrix the columns of which are successive first difference vectors u.
U^+	Generalized inverse of U
u	First difference vector
$\tilde{u}$	First difference vector resulting from putting extrapolated vector $\tilde{x}$ back into iteration

u_0	Initial first difference vector of a sequence
x	Generic vector in m-space
x_0	Initial vector in a sequence
x_1	First output vector of iteration Equation (1.1)
x_2	Second output vector
$\tilde{x}$	Extrapolated vector (see Equations (3.3) and (4.6))
$\hat{x}$	Exact solution vector (see Equations (2.7) and (3.2))
ε	Tolerance to be met by norm of R
λ	Eigenvalue of A
ξ	Auxiliary vector in Equations (3.3), (3.4), (4.6), (4.7)

FORTRAN PARAMETERS

ISK	Number of iterations skipped over before beginning an extrapolation cycle
M	Same as m above
MFB	Controls whether output of $\tilde{x}$ of an extrapolation cycle, for which tolerance is not met, is or is not forced back as input to next iteration
MON	Controls whether input to an iteration is only the previous vector XOLD, or whether components of the output vector XNEW are fed in as soon as they have been generated
MXT	Maximum allowed value of iteration counter NIT (to prevent runaway computations)
N	Assigned maximum number of columns of H, also maximum allowed rank of extrapolation
NC	Number of columns } in rectangular grid of sample problem
NL	Number of layers } in rectangular grid of sample problem
NR	Number of rows } in rectangular grid of sample problem
NIT	Iteration counter

NSK	Number of iterations skipped between end of one extrapolation cycle and beginning of the next
TOL	Same as ϵ above

FORTRAN ARRAYS AND DIMENSIONS

DIF1 (M,N+1)	Same as U
DIF2 (M,N)	Same as H
HUTF (N,N+1)	Holds (H*H,$H^{+}u_0$) and also the results of triangularization thereof
RSID (M)	Same as R
XHLD (M)	Holds initial vector x_0 through an extrapolation cycle
XKSI (N)	Same as ξ
XNEW (M)	Output of iteration, x_{n+1}
XOLD (M)	Input of iteration, x_n

ABSTRACT

A new family of methods, called reduced rank extrapolation, is developed for accelerating convergence of the sequence of vectors generated during the iterative solution of a system of m linear algebraic equations in m unknowns. Large systems of this kind arise, for example, in the finite difference or finite element solution of partial differential equations.

Reduced rank extrapolation is derived from full rank extrapolation, which is a straightforward generalization to vector space of the well known Aitken Δ^2-Shanks e_1, scalar extrapolation. It is applicable when the iteration has reached a point where only a few, say r, eigenvalues dominate the situation and hence only r difference vectors can be linearly independent to specified tolerance. The rank, r, is determined during the solution of an auxiliary problem of best approximation in vector space, i.e., "best" in the sense of minimizing some specified vector norm. Least squares theory, corresponding to the Euclidean norm, is developed in detail herein.

Application to Laplace's equation in a square and in a cube yielded reduction in computation time by a factor ranging between 2.4 and 4.7, and reduction in iteration count by a factor ranging between 3.6 and 5.4.

ADMINISTRATIVE INFORMATION

This research was carried out under the NAVSEA Mathematical Sciences Research Program, Task Area SR0140301, Element 61153N, Work Unit 1808-010.

1. INTRODUCTION AND SUMMARY

This report presents a new family of methods, called reduced rank extrapolation, for accelerating or producing convergence of the sequence of vectors generated in the iterative solution, by whatever scheme, of a system of linear algebraic equations in m variables

$$x_{n+1} = A\, x_n + b \tag{1.1}$$

where x and b are $m \times 1$ vectors and A is a (constant) $m \times m$ matrix. Large systems of this type occur in the finite difference or finite element solution of partial

differential equations of elliptic or parabolic type. These arise in many important physical and engineering problems, such as those concerned with the flow of water around ship hulls, the flow of air around aircraft and missiles, and the flux of neutrons in a nuclear reactor. Standard methods of setting up the iteration matrix A in terms of the original background equations are the subject of such well known textbooks as those of Varga (1962),[1*] Wachspress (1966),[2] Young (1971),[3] and Forsythe and Moler (1967).[4]

The motivating idea behind the reduced rank methods can be developed as follows. In the background is the well known formula for extrapolating to the limit of a scalar sequence

$$\tilde{x} = (x_{n-1}x_{n+1}-x_n^2)/(x_{n+1}-2x_n+x_{n-1})$$

$$= x_{n-1} - (x_n-x_{n-1})^2/(x_{n+1}-2x_n+x_{n-1}) \qquad (1.2)$$

due to Aitken (1926)[5] and to Shanks (1949),[6] (1955).[7] One derivation of this formula can be generalized in a straightforward manner to apply to vector sequences. The result, Equation (3.2), is here called full rank extrapolation; it seems to be known, but not "well known." (The treatment in Section 3 is taken from the present author's working notes dated 1952.) Full rank extrapolation can be helpful in the iterative solution of small, mildly nonlinear systems. However, it is of no use for large linear systems because it requires the inversion of an $m \times m$ matrix of second difference vectors. This matrix has the same size as the original system and is very likely to be much more ill conditioned.

This situation invites exploration of how much can be accomplished with only a few difference vectors--say r of them, where $1 \leq r \ll m$. (This is the origin of the term "reduced rank extrapolation".) The outlook is hopeful when one recalls that, as the iteration of Equation (1.1) proceeds, the successive x_n become expressible, to a given tolerance, in terms of fewer and fewer eigenvectors of A--i.e., those associated with the eigenvalues of greatest magnitude. Under these conditions, only a corresponding number of first difference vectors or second difference vectors can

*A complete listing of references is given on page 45.

be linearly independent--again to specified tolerance. It is not known in advance just how many such vectors will be needed; the number is a function of the tolerance imposed and is determined in the course of computation.

The theory of reduced rank extrapolation, as developed in Section 4, is both simple and elegant. It differs from most other methods of accelerating vector sequences in two respects: (1) vectors themselves, not their individual components, are regarded as the basic entities, (2) only "observables," i.e., readily computable quantities such as iteration vectors, difference vectors, and vector norms appear in the final version of the theory. Although eigenvalues lurk in the background, there is no question of estimating them and there are no difficulties associated with close, repeated, or degenerate eigenvalues.

Reduced rank extrapolation can be described as follows: let x_0 be some selected vector in the sequence generated by the basic iteration of Equation (1.1), let $u_0 = x_1 - x_0$, and let $\tilde{x}$ be the result of rank r extrapolation. Then $\tilde{x} - x_0$ is expressed as a linear combination of r successive first differences u_0, u_1, ... u_{r-1}, the coefficients being those by which $(-u_0)$ is "best" represented in terms of r successive second difference vectors h_0, h_1, ... h_{r-1}. The three interpretations of "best approximation" in terms of the standard vector norms ℓ_1, ℓ_2, and ℓ_∞, combined with the several computational schemes available for optimization under each of these norms and the flexibility in r, give rise to the family of extrapolation methods announced in the title.

There are two types of limitations on the use of reduced rank extrapolation. The first is inescapable: (I-A) must be nonsingular so that none of the eigenvalues of A can be +1. In fact, if any one of them is too close to +1, the extrapolated vector may be seriously in error. This comment is made precise by the error bound, Equation (4.12). The second limitation has to do with available computer storage capacity for the arrays of first and second difference vectors required. This matter will be taken up in Section 5.

2. PRELIMINARIES ON DIFFERENCES AND ERRORS

From the vector iteration Equation (1.1) it is trivial to show that the successive m-component first difference vectors

$$u_n = x_{n+1} - x_n \tag{2.1}$$

satisfy

$$u_n = A\, u_{n-1} = \ldots = A^n\, u_0 \tag{2.2}$$

The second difference vectors are given by

$$h_n = u_{n+1} - u_n = (A-I)\, u_n \tag{2.3}$$

A central role in the theory to be developed here is played by the following $m \times r$ $(1 \leq r \leq m)$ rectangular matrices whose columns are first or second difference vectors, as indicated:

$$U = (u_0, u_1, \ldots u_{r-1}) \tag{2.4}$$

$$H = (h_0, h_1, \ldots h_{r-1}) \tag{2.5}$$

It follows from Equation (2.3) that

$$H = (A-I)\, U \tag{2.6}$$

This relation plays a key role in the theory.

Error vectors behave much like difference vectors. Let $\hat{x}$ be the exact solution

$$\hat{x} = A\, \hat{x} + b \tag{2.7}$$

Then it is trivial to show that

$$x_n - \hat{x} = A(x_{u-1} - \hat{x}) = \ldots = A^n(x_0 - \hat{x}) \tag{2.8}$$

There is actually an intimate relationship between a difference vector and the corresponding error vector:

$$u_n = x_{n+1} - x_n = A\,x_n + b - x_n = (A-I)\,x_n - (A-I)\,\hat{x}$$

$$= (A-I)\,(x_n - \hat{x}) \tag{2.9}$$

analogous to the relation in Equation (2.3) between second and first order difference vectors.

3. FULL RANK EXTRAPOLATION

Let $\{x_n\}$ be the sequence of vectors generated by Equation (1.1). Then formally

$$x_n = x_0 + (x_1 - x_0) + (x_2 - x_1) + \ldots + (x_n - x_{n-1})$$

$$= x_0 + u_0 + u_1 + \ldots + u_{n-1}$$

$$= x_0 + (I + A + A^2 + \ldots + A^{n-1})\,u_0$$

$$= x_0 + (I-A)^{-1}\,(I-A^n)\,u_0$$

If all eigenvalues of A are less than unity in absolute value, then $\lim_{n\to\infty} A^n = 0$, and the vector sequence converges to the solution:

$$\lim_{n\to\infty} x_n = \hat{x} = x_0 + (I-A)^{-1}\,u_0 \tag{3.1}$$

Actually it is trivial to prove, by substitution into Equation (2.7), that the second equality here is an identity; it holds regardless of convergence provided only that the indicated inverse exists, i.e., that no eigenvalue of A can be +1.

If the matrices U and H are both $m \times m$, and if H is nonsingular, then from Equation (2.6) it follows that $(I-A)^{-1} = -\,U\,H^{-1}$ and Equation (3.1) becomes

$$\hat{x} = x_0 - U\,H^{-1}\,u_0 \tag{3.2}$$

which can also be written as the pair of equations

$$\hat{x} = x_0 + U\,\xi \tag{3.3}$$

$$0 = u_0 + H\,\xi \tag{3.4}$$

In either of these forms this vector extrapolation formula constitutes the full rank generalization to m-dimensional vector space of the well known Aitken Δ^2-Shanks e_1 formula, Equation (1.2), for extrapolating scalar sequences: both Equations (3.2) and (1.2) are derived by essentially identical chains of reasoning and Equation (3.2) reduces to Equation (1.2) for m = 1.

Although theoretically Equation (3.2) yields the exact limit, $\hat{x}$, it is not really useful for solving linear systems because it requires solving a linear system of the same order as the original. However, it can be useful in the iterative solution of mildly nonlinear systems and it does provide a pattern for the more useful reduced rank extrapolation.

4. REDUCED RANK EXTRAPOLATION DEFINITION AND JUSTIFICATION

As originally conceived, the term "reduced rank extrapolation" was intended to convey the idea that the rank of the matrices U and H appearing in the extrapolation formulas of Equations (3.2), (3.3), and (3.4) is not m, the dimension of the vector space, but rather some smaller integer $r{:}1 \leq r < m$. In other words, only r successive first difference vectors $u_0, u_1, \ldots u_{r-1}$, or second difference vectors $h_0, h_1, \ldots h_{r-1}$, are linearly independent, at least to the accuracy carried in computation. This situation obtains when the iteration of Equation (1.1) has progressed sufficiently far that contributions associated with the m-r eigenvalues smallest in magnitude are no longer significant.

The proof of this last statement involves the reduction of the iteration matrix A to classical (Jordan) canonical form J by means of a similarity transformation

$$A\,W = W\,J \tag{4.1}$$

The columns of W are the eigenvectors (or principal vectors) of A, and J is a diagonal matrix of eigenvalues (perhaps with Jordan boxes) arranged in order of decreasing magnitude. The eigenvalues must all satisfy

$$|\lambda| < 1 \tag{4.2}$$

if the original iteration process is to converge at all. From Equation (4.1) it follows that

$$A^n W = W J^n \tag{4.3}$$

According to the preceding discussion, if n is large enough, then all elements in some lower right quadrant of J^n will be smaller in magnitude than any preassigned threshold value. Correspondingly, all main diagonal elements of the r × r upper left quadrant will have magnitudes above the threshold and will be considered significant. Now let u_0 be expressed as a linear combination of the eigenvectors of A:

$$u_0 = W y_0 \tag{4.4}$$

The corresponding expression for u_n is obtained by using Equations (2.2) and (4.3):

$$u_n = A^n u_0 = A^n W y_0 = W J^n y_0 = W y_n \tag{4.5}$$

As a result of the properties of J^n already described, only the first r components of y_n will be of significant magnitude; thus, u_n lies in a subspace of dimension r and only r consecutive u's can be linearly independent. Actually, this is an idealized situation; practically, the distinction between rank r and rank r + 1 may be very fuzzy indeed. Ways of dealing with this troublesome matter will be discussed later in this report.

BASIC EQUATIONS

In the full rank extrapolation discussed in Section 3, it was shown that the difference between the exact solution, $\hat{x}$, and some *initial vector*, x_0, is a linear combination of the m columns of U. These, being linearly independent by hypothesis, span the whole m-dimensional vector space. This simple property of full rank extrapolation is carried over as closely as possible to reduced rank extrapolation: the difference between the extrapolated vector, $\tilde{x}$, and the initial vector, x_0, is a linear combination of the r columns of U which, by assumption, are linearly independent. Thus, Equation (3.3) is retained

$$\tilde{x} = x_0 + U\,\xi \tag{4.6}$$

with the change that the vector ξ now has only r components ($1 \leq r \leq m$), and the left side is merely an extrapolated vector, $\tilde{x}$, not the exact solution, $\hat{x}$. However, the companion Equation (3.4) is no longer available, as it stands, for the determination of ξ because $(I-A)^{-1}$ can no longer be completely determined from $H = (A-I)\,U$ as it was in full rank extrapolation. Instead, a new vector,

$$R = u_0 + H\,\xi \tag{4.7}$$

called the residual vector, is introduced, and ξ is chosen to minimize some norm of R. Before the latter point is elaborated, it will be shown that R has a natural and important interpretation as the first difference, $\tilde{u}$, obtained by substituting the extrapolated vector, $\tilde{x}$, back into the basic iteration of Equation (1.1):

$$R = u_0 + H\,\xi = (Ax_0+b-x_0) + (A-I)\,U\,\xi$$

$$= (A-I)\,(x_0+U\xi) + b = (A\tilde{x}+b) - \tilde{x} = \tilde{u} \tag{4.8}$$

Furthermore, R is intimately related to the error, $\tilde{x} - \hat{x}$, of the extrapolated vector, just as u_0 is related to the initial error, $x_0 - \hat{x}$. As for the latter, merely

rearranging Equation (3.1) gives

$$u_0 = (A-I)\ (x_0-\hat{x}) \tag{4.9}$$

Then it follows easily that

$$\begin{aligned} R = \tilde{u} = u_0 + H\ \xi &= (A-I)\ (x_0-\hat{x}) + (A-I)\ U\ \xi \\ &= (A-I)\ (x_0-\hat{x}+\tilde{x}-x_0) \\ &= (A-I)\ (\tilde{x}-\hat{x}) \end{aligned} \tag{4.10}$$

Both of these equations correspond to Equation (2.9).

VECTOR NORMS AND THE DETERMINATION OF ξ AND r

Under the conditions assumed in this section it is no longer possible to make the residual vector, R, vanish as it did in the full rank case. The best that can be done is to make it as "small" as possible by choosing the vector ξ so as to minimize some norm of R and make this value less than some prespecified tolerance:

$$||R|| = \min \leq \varepsilon \tag{4.11}$$

There are three standard vector norms, ℓ_1, ℓ_2, and ℓ_∞, which can be used to measure the magnitude of a vector. Corresponding to each of these vector norms is a matrix norm which is "consistent" with it. This property of consistency, when, applied to Equation (4.10), asserts that

$$||\tilde{x}-\hat{x}|| \leq ||(A-I)^{-1}||\ ||R|| \tag{4.12}$$

These norms, applied to a vector R with components R^i $(1 \leq i \leq m)$ and to a matrix $M = (A-I)^{-1}$ with components M_{ij}, are:

$$\ell_1: \quad ||R||_1 = \sum_{i=1}^{m} |R^i|$$

$$||M||_1 = \max_j \sum_{i=1}^{m} |M_{ij}| \tag{4.13}$$

$$\ell_2: \quad ||R||_2^2 = R^*R = \sum_{i=1}^{m} |R^i|^2$$

$$||M||_2^2 = \lambda_1 = \text{largest eigenvalue of } M^*M \tag{4.14}$$

$$\ell_\infty: \quad ||R||_\infty = \max_i |R^i|$$

$$||M||_\infty = \max_i \sum_{j=1}^{m} |M_{ij}| \tag{4.15}$$

Among these norms, ℓ_2 (better known as the Euclidean norm) is by far the best known and widely used. Optimization using it is the method of least squares introduced by Legendre and by Gauss around 1800. The vector ξ is determined by solving a system of linear equations of order r, called the normal equation.

The next best known norm is the ℓ_∞ norm. Optimization using it is known as Chebyshev or minimax approximation. The vector ξ is determined as the solution of a linear programming problem.

The least well known of these norms is the ℓ_1 norm, which seems to have no other name. Optimization using it is sometimes referred to as the method of least absolute deviations. The vector ξ is determined as the solution of a linear programming problem in this case also.

Thus far the three vector norms ℓ_1, ℓ_2, and ℓ_∞ have been treated on an equal footing. Actually, all of the author's computational experimentation has been with the ℓ_2 norm, simply because it was more familiar and hence easier to adapt to present needs. Similar adaptation of linear programming methods encountered in optimization under ℓ_1 and ℓ_∞ norms should be carried out. For a sample of fairly recent work in ℓ_1 and ℓ_∞ optimization, see the several papers by Barrodale et al.[8-10] and by Bartels et al.[11-14]

COMPUTATIONAL TACTICS

Determination of the rank, r, of the system

$$R = u_0 + H\,\xi \tag{4.7}$$

$$||R|| = \min \overset{<}{-} \varepsilon \tag{4.11}$$

is a key step in the solution of the extrapolation problem. Considerable computational experimentation has convinced the author that the best approach is to begin by letting the basic iteration, Equation (1.1), run ("initial skip") until contributions associated with smaller eigenvalues have been suppressed as described earlier under "Definition and Justification." At this stage the effective rank of H will be some not-too-large integer. Then begin a "build up" cycle, i.e., carry out a step by step procedure in which the output of each new iteration of Equation (1.1) is used to build a new column of each of the difference matrices U and H. As each new column is added, the vector ξ is determined so as to minimize whatever norm of R is being used. If the resulting minimized norm is less than some preassigned tolerance, ε, then the solution is at hand and the extrapolated solution vector is obtained from

$$\tilde{x} = x_0 + U\,\xi \tag{4.6}$$

using the current x_0, U, and ξ. The number of columns of U and of H is r, the rank of the system. If the tolerance is not met, the iteration cycle is repeated until the number of columns reaches some maximum number, N, determined in advance by

availability of storage for U and H. Then a new build-up cycle can be started either immediately or after skipping through some preassigned number (NSK) of basic iterations of Equation (1.1).

It is a matter of tactical doctrine whether or not to feed back, as a new starting vector, an extrapolated vector $\tilde{x}$ for which the tolerance on the norm of R has not been met. Computational evidence, which appears in Section 5, suggests that this is a minor question, provided that a sufficiently large maximum rank is allowed for (N>2) and that the initial skip is sufficient for the effective rank of H to have become $\leq$ N as required by hypothesis. The very worst performance arises from forced feedback of rank one extrapolation right from the beginning (N=1, ISK=0).

From the foregoing discussion it should be clear that what has been called the rank of the system (number of columns of U and H, number of components of ξ) depends not only on the course of the basic iteration of Equation (1.1), i.e., how rapidly it converges, but also on the tolerance ε which is imposed. An analogous situation, wherein the solution of a small system of linear equations depends strongly on the tolerance imposed on the residuals, was discussed by Peters and Wilkinson (1970),[15] (especially, pages 314-315).

LEAST SQUARES OPTIMIZATION STANDARD THEORY PLUS MODIFICATIONS

The rest of this section will be devoted to ℓ_2 optimization, the well known method of least squares. The vector ξ in the optimization problem given by Equations (4.7), (4.11) is found as the solution of the so-called normal equations

$$H^* u_0 + H^*H \xi = 0 \tag{4.16}$$

and is given by

$$\xi = - H^+ u_0 \tag{4.17}$$

where

$$H^+ = (H^*H)^{-1} H^* \tag{4.18}$$

is the well known Moore-Penrose generalized inverse of H. (General references: Ben-Israel and Greville (1974),[16] Boullion and Odell (1971),[17] and Lawson and Hanson (1974).[18]

In view of the tentative step-by-step procedure advocated in the preceding section, one might be inclined to use a step-by-step expansion of H^+ using partitioned matrices as was so carefully discussed by Greville (1960).[19] However, from the standpoint of programming the solution on a computer, it is much simpler and more elegant to use what may be called the expanding Choleski method. This method involves the triangular square root, T, of the symmetric positive definite matrix H*H. From

$$T^*T = H^*H \tag{4.19}$$

it follows that

$$\left.\begin{aligned} T_{11} &= \sqrt{(H^*H)_{11}} \\ T_{1j} &= (H^*H)_{1j}/T_{11} \\ T_{ij} &= \left((H^*H)_{ij} - \sum_{h=1}^{i-1} T_{hi}\, T_{hj}\right)\Big/ T_{ii} \quad (i<j) \\ T_{ii} &= \sqrt{(H^*H)_{ii} - \sum_{h=1}^{i-1} T_{hi}^2} \end{aligned}\right\} \tag{4.20}$$

The process of constructing T produces from Equation (4.16) the triangular system

$$T\,\xi + T^{*-1}\, H^*\, u_0 = 0 \tag{4.21}$$

which is easily solved for ξ.

The expanding nature of the computation carries over very neatly into the coding as follows: as each new vector (beyond x_0, x_1, x_2) is produced by the basic iteration, a new column is added to the difference matrices U and H. Correspondingly, a new row and a new column are added to H*H and a new element (new row) is added to H* u_0. The lower triangular matrix T gains a new column and the vector T^{*-1} H* u_0 in Equation (4.21) gains one more element. Once any element of the array holding the system of Equation (4.16) has been used in computation, it is never needed again; hence the same array can be overlayed by the system of Equation (4.21). The expansion process may have to be halted prematurely if some $(T_{ii})^2$ becomes either too small in relation to overall accuracy of computation or actually negative. In such a case it is best to just abort the current extrapolation cycle and return control to the executive routine.

Geometric Sidelights

Use of the ℓ_2 norm leads to especially simple relations between the residual vector R = $\tilde{u}$, Equation (4.7), and u_0 and also between the errors of the extrapolated vector $\tilde{x}$ and of the initial vector x_0. Substituting Equation (4.17) into Equation (4.8) gives

$$R = \tilde{u} = u_0 + H\,\xi = (I-HH^+)\,u_0 \tag{4.22}$$

The quantity $(I-HH^+)$ is called an orthogonal projector: it eliminates any components of u_0 which lie in the subspace spanned by the columns of H.

Similarly, Equations (4.9) and (4.10) with Equation (4.22) yield

$$\tilde{x} - \hat{x} = (A-I)^{-1}\,\tilde{u} = (A-I)\,(I-HH^+)\,(A-I)\,(x_0-\hat{x})$$

$$= (I-UU^+)\,(x_0-\hat{x}) \tag{4.23}$$

The last step involves the basic relation

$$H = (A-I)\,U \tag{2.6}$$

plus a theorem of Greville (1966)[20] which justifies assuming that

$$H^{+} = U^{+} (A-I)^{-1} \tag{4.24}$$

Here again, in Equation (4.23), is an orthogonal projector which eliminates any components of the initial error vector $x_0 - \hat{x}$ which lie in the subspace spanned by the columns of U.

Obviously, these projectors are related by a similarity transformation:

$$I - HH^{+} + (A-I)\ (I-UU^{+})\ (A-I)^{-1} \tag{4.25}$$

Special Cases: N = 1 and 2

The simplest possible cases of the foregoing theory occur at the start of an extrapolation cycle when the vector difference matrices U and H have only one or two columns (N=1 or 2). Explicit solutions are easily given, at least for the ℓ_2 norm:

$$N = 1 \quad \xi = - h_0^{*}u_0 / h_0^{+}h_0 \ \text{(scalar)} \tag{4.26}$$

$$\tilde{x} = x_0 + \xi\ u_0 = (1-\xi)\ x_0 + \xi\ x_1 \tag{4.27}$$

Thus the rank one extrapolation is merely the well known relaxation process with a ready-made value of the relaxation parameter. Corresponding expressions for N = 2 are

$$\tilde{x} = (1-\xi_1)\ x_0 + (\xi_1-\xi_2)\ x_1 + \xi_2\ x_2 \tag{4.28}$$

where

$$\left.\begin{aligned} \xi_1 &= - \frac{(h_1^{*}h_1)\ (h_0^{*}u_0) - (h_1^{*}h_0)\ (h_1^{*}u_0)}{(h_1^{*}h_1)\ (h_0^{*}h_0) - (h_1^{*}h_0)\ (h_0^{*}h_1)} \\ \xi_2 &= - \frac{-(h_0^{*}h_1)\ (h_0^{*}u_0) + (h_0^{*}h_0)\ (h_1^{*}u_0)}{(h_1^{*}h_1)\ (h_0^{*}h_0) - (h_1^{*}h_0)\ (h_0^{*}h_1)} \end{aligned}\right\} \tag{4.29}$$

As might be expected, these simplest cases have already appeared in the literature. For example, Jennings (1971)[21] gave the equivalent of Equation (4.26) and called it SDM (second difference modulation). He also discussed what he called "double acceleration" which appears to be closely related to the N = 2 case of the present report, as did Hadjidimos (1978).[22]

Earlier Version: Singular Value Decomposition

For the sake of completeness, it is desirable to describe briefly an earlier version of reduced rank extrapolation which has already appeared in the literature (Eddy (1979))[23] and to explain why it has been discarded.

The original train of thought, as described at the beginning of Section 4, led to attempts to determine "the rank" of the matrix H of second difference vectors and then to solve the normal Equations (4.16) having this dimension. The most powerful tool for such rank determination and for solving the linear least squares problem is the "singular value decomposition"

$$H = W\ D\ V^* \tag{4.30}$$

where D is a diagonal matrix of (nonnegative) eigenvalues of H*H and V and W are orthogonal matrices. (General references: Ben-Israel and Greville (1974),[16] Stewart (1973)).[24] The computational aspects of singular value decomposition were perfected by Golub and various collaborators in a series of papers during the 1960's, (e.g., Golub and Reinsch (1970))[25] culminating in a state-of-the-art computer program, CSVD, (in FORTRAN) for general complex matrices (Businger and Golub (1969)).[26] This program was adapted by the present author to the simpler case of real arithmetic only (Computer Program RSVD) and used for computational experimentation as reported in Eddy (1979).[23]

The dismaying fact which emerged from monitoring printouts was that, at least for the typical sample problem described in Section 5, there was no clear-cut break between "large" and "small" singular values of H, analogous to assumptions explained at the beginning of this section, and hence no clear-cut rank for H.

Correspondingly, it became clear that linear independence of the vectors encountered here is a rather "fuzzy" concept: the number of successive vectors which can be considered to be linearly independent depends very much on the tolerance imposed.

Another fact emerged serendipitously from the copious printouts, something that had not even been conjectured previously, namely that

$$R = \tilde{u} \tag{4.8}$$

Once conjectured, this equality was easily proved and became the cornerstone of the revised approach described herein.

5. COMPUTATIONAL EXPERIMENTS

SAMPLE PROBLEM

The sample problem used to exercise the computer program and to compare the results of various tactical options therein is the finite difference solution of the Laplace equation, in both a square (two dimensions) and a cube (three dimensions) with zero boundary conditions. Ten internal mesh points are taken in each direction, and mesh point starting values are obtained from a pseudo-random number generator. This generator is a function subroutine built into the CDC FORTRAN compiler. It produces the same sequence of "random" numbers on each problem run. This simple problem is intended to be sufficiently realistic but at the same time to be easy to code.

COMPUTER PROGRAM

The computer program is intended to simulate some existing computer program for vector iteration into which an extrapolation capability has been inserted. Furthermore, it is also intended that the program be easy to use for numerical experiments so certain tactical options are made to depend upon the values of input parameters.

In shifting between two- and three-dimensional problems, it is necessary to replace the ordinary iteration subroutine, to change most of the dimension statements, and to change the format statement controlling the page heading for output printouts.

The program is organized into three modules for which listings are given in the appendix:

(1) the executive routine, SIMP, which controls initial setup, controls printouts for monitoring, calls the ordinary iteration subroutine, computes norms of difference vector and tests for convergence, and calls the extrapolation subroutine.

(2) the ordinary iteration subroutine, P2D or P3D, depending on whether a two- or a three-dimensional problem is at hand. In either case the new value at a mesh point is just the mean of the values at nearest neighbor points. This averaging is under control of an input parameter, MON: for MON = 0 only old mesh values (components of XOLD) are used in computing new values (gesamtschritt = total step = point Jacobi iteration); for MON = 1 new mesh point values (components of XNEW) are fed in as soon as they are available (einzelschritt = single step = Gauss-Seidel iteration). No effort was made to optimize these subroutines in any sense.

(3) the extrapolation subroutine, XTRP, which builds up the difference matrices DIF1 (=U) and DIF2 (=H), stores the vector x_0 at the beginning of a build-up cycle in XHLD, builds up the normal equations and their triangularized version (both in the array HUTF), calculates the residual vector RSID and its norms under ℓ_1, ℓ_2, and ℓ_∞, tests whether a selected one of these norms meets the specified tolerance TOL (=ε), and, if so, computes the extrapolated vector $\tilde{x}$ which then replaces the least iteration output in XNEW. This $\tilde{x}$ is then fed back into the ordinary iteration; since $\tilde{u}$ = R, the convergence test there is met and computation is stopped.

There are several more input parameters:

M = dimension of original system of equations = number of components in most vectors. For the present sample problem, M = NR × NC for 2D problems, M = NR × NC × NL for 3D problems where

NR = number of rows

NC = number of columns

NL = number of layers

in the rectangular array of mesh points. All were equal to ten in the sample problems.

N = maximum number of columns allowed for in DIF2 (=H) = maximum rank extrapolation allowed for. This is also the maximum number of components in the auxiliary vector XKSI and the maximum number of rows in the normal equation array HUTF. Values of N from one to ten were tried.

MXT = maximum [illegible] this is a safety feature to prevent [illegible]

ISK = initial skip [illegible] iterations skipped through before an extrapolation build-up [illegible]

NSK = number of [illegible] skipped through between the end of one extrapolation build-up cycle and the beginning of the next.

MFB controls forced feedback into the iteration cycle of an extrapolated vector $\tilde{x}$ for which the convergence criterion $R < \varepsilon$ has not been met. (No feedback for MFB=0, forced feedback for MFB=1).

Dimensions of all arrays are specified in terms of the two parameters M and N. Vectors XOLD, XNEW, XHLD, and RSID have dimension M. Vector XKSI has dimension N. The two-dimensional arrays are DIF1 (M,N+1), DIF2 (M,N), and HUTF (N,N+1).

The amount of storage required for these arrays is

$$S = (5+2N)\ M + N\ (N+2) \tag{5.1}$$

For M of the order of thousands or tens of thousands, as would be the case for a realistic problem in partial differential equations, this amount of extra storage becomes prohibitively large for present day computer systems, even for small values of N. (N = 5 or 6 seems to be optimal for the present sample problem.)

SUMMARY OF RESULTS

Tables 1-3 show computation time in seconds on a CDC 6400 and iteration count as functions of two parameters: N, the maximum rank extrapolation, and ISK, the numbers of iterations skipped through before beginning the first extrapolation cycle. No complete survey was attempted--only enough to locate the parameter values which yielded minimum computation times.

Three closely related problems were treated: the finite difference equivalents of Laplace's equation in

(1) 2 dimensions, 10 × 10 grid, 5-point pattern

(2) 2 dimensions, 10 × 10 grid, 9-point pattern

(3) 3 dimensions, 10 × 10 × 10 grid, 7-point pattern

For each of these problems there were four combinations of tactical options, not all of which were investigated for each problem. They were

TABLE 1 - COMPUTATION TIMES AND ITERATION COUNTS 2D, 5-POINT PATTERN

TABLE 1A - MON=0, MFB=0

ISK \ N	1	2	3	4	5	6	7	8	9	10
0	6.449	4.449	3.815	3.266	2.969	3.296	3.527	3.565	3.710	3.701
	239	139	105	81	67	70	70	68	66	67
.										
.										
.										
50					1.610					
					69					
55				1.888	1.421	1.491				
				81	68	69				
60			2.538	1.770	1.234	1.211	1.229			
			105	81	67	67*	67			
65				1.640	1.298	1.306				
				81	72	72				
70				1.508	1.382					
				81	77					
75				1.383						
				81						

TABLE 1 (Continued)

TABLE 1B - MON=0, MFB=1

N / ISK	1	2	3	4	5	6	7	8	9	10
0	14.008	6.469	4.721	4.077	3.628	3.237	3.377	3.532	3.412	3.210
	476	188	122	95	78	66	63	62	57	51
.										
.										
.										
25				2.291	1.931	1.852	1.653	1.584	1.387	1.434
				70	59	59	49	47	43	43
30	10.647	2.947	2.903	1.995	1.619	1.487	1.287	1.377	1.337	1.506
	378	102	93	66	55	51	46	47	46	48
35				1.801	1.421	1.199	1.226	1.313	1.376	1.372
				65	54	49	49	50	51	51
40	11.922	2.211	1.915	1.608	1.213	1.158	1.192			
	426	86	74	64	53	52	52			
45				1.476	1.165	1.188	1.216			
				64	56	56	56			
50	7.645	1.577	1.748	1.445	1.144	1.271	1.340			
	285	74	76	66	59*	61	62			
55		1.512	1.564	1.304	1.203	1.281				
		74	74	66	64	65				
60	3.922	1.464	1.366	1.262	1.231	1.229				
	163	76	72	69	67	67				
65		1.291	1.406	1.297						
		74	76	73						
70	2.510	1.472	1.369							
	119	82	78							

TABLE 1 (Continued)

TABLE 1C - MON=1, MFB=0

ISK \ N	1	2	3	4	5	6	7	8	9	10
0	1.564	1.298	1.220	1.428	1.383	1.340	1.321	1.399	1.583	1.719
	57	40	33	35	31	28	25	26	28	29
.										
.										
.										
20				0.915	0.845	0.711	0.708			
				35	32	28	28			
25			0.725	0.795	0.708	0.709				
			34	35	32	32				
30		0.804	0.673	0.662	0.662					
		40	35	35*	35					
35	1.176	0.689	0.683	0.692						
	58	39	39	39						
40	1.099	0.765								
	57	44								

TABLE 1 (Continued)

TABLE 1D ~ MON=1, MFB=1

ISK \ N	1	2	3	4	5	6	7	8	9	10
0	4.052	2.033	1.931	2.021	1.895	1.983	2.224	2.144	2.113	2.010
	138	58	49	47	41	40	41	37	36	32
5	2.980	1.739	1.634	1.841	1.534	1.675	1.659	1.913		
	103	53	45	46	36	38	35	38		
10	2.043	1.451	1.296	1.374	1.128	1.256	1.069	1.073		
	73	48	39	39	32	32	27	27		
15	2.357	1.116	1.005	0.867	0.838	0.783	0.722	0.799	0.805	0.780
	87	40	35	30	28	27	25	25	25	25
20	1.673	0.929	0.666	0.677	0.655	0.707				
	66	38	29	29	28*	28				
25	1.252	0.742	0.690	0.655	0.717	0.713				
	54	35	33	32	32	32				
30	0.933	0.712	0.662	0.654	0.667	0.658				
	46	37	35	35	35	35				
35	0.814	0.685	0.675							
	44	39	39							
40	0.832	0.757								
	47	44								

TABLE 1 (Continued)

TABLE 1E - SUMMARY FOR 2D, 5-POINT PATTERN
OPTIMAL VALUES OF PARAMETERS

	Free Run	Extrapolation	
Iteration Input Vector		No Forced Feedback MFB=0	Forced Feedback MFB=1
XOLD ONLY MON=0	Time = 4.803 NIT = 319	Min Time = 1.211 NIT = 67 ISK = 60 N = 6 Time Gain = 3.97 NIT Gain = 4.76	Min Time = 1.144 NIT = 59 ISK = 50 N = 5 Time Gain = 4.20 NIT Gain = 5.41
XOLD/XNEW (as available) MON=1	Time = 1.943 NIT = 128	Min Time = 0.662 NIT = 35 ISK = 30 N = 4 Time Gain = 2.94 NIT Gain = 3.66	Min Time = 0.655 NIT = 28 ISK = 20 N = 5 Time Gain = 2.97 NIT Gain = 4.57

TABLE 2 - COMPUTATION TIMES AND ITERATION COUNTS 2D, 9-POINT PATTERN

TABLE 2A - MON=0, MFB=0

ISK \ N	1	2	3	4	5	6	7	8	9	10
0	2.718	2.047	2.056	1.988	2.088					
	73	46	41	36	36					
.										
.										
.										
20	2.408	1.601	1.514	1.364	1.526					
	73	45	41	36	38					
30	2.259	1.422	1.175	1.080	1.071					
	73	46	39	36	36*					
40	2.098	1.263	1.191	1.190						
	73	47	45	45						
50	1.951									
	73									

TABLE 2 (Continued)

TABLE 2B - MON=0, MFB=1

ISK \ N	1	2	3	4	5	6	7	8	9	10
0	14.126	5.926	4.126	3.403	3.729		2.805		2.866	
	252	105	70	55	49		39		37	
5										
10	12.499	4.627	3.180	2.512	2.437		1.949		1.446	
	230	88	60	47	44		34		26	
15					1.613					
					34					
20	8.278	2.771	1.568	1.239	1.183		1.298		1.240	
	160	61	39	33	31		32		30	
25	6.515	1.545	1.090	1.198	1.167					
	131	42	34	35	34					
30	4.357	1.102	1.095	1.064	1.067		1.060		1.074	
	96	37	37	36*	36		36		36	
35	2.427	1.144	1.103	1.096	1.094					
	64	41	40	40	40					
40	1.510	1.159	1.197	1.206	1.192		1.195		1.184	
	51	45	45	45	45		45		45	
45										
50	1.388	1.337	1.337	1.332	1.331		1.329		1.335	
	55	54	54	54	54		54		54	

TABLE 2 (Continued)

TABLE 2C - MON=1, MFB=0

ISK \ N	1	2	3	4	5	6	7	8	9	10
0	1.526	1.295	1.316	1.413						
	39	28	25	25						
5										
10	1.383	1.158	1.156	1.104						
	39	29	27	25						
15	1.342	1.006	1.055	0.946						
	40	28	28	25						
20	1.227	0.977	0.790	0.813						
	39	30	25*	25						
25	1.190	0.839	0.828	0.836						
	40	29	29	29						
30	1.075	0.939	0.940	0.946						
	39	34	34	34						
35										
40	1.077	1.104								
	43	43								

TABLE 2 (Continued)

TABLE 2D - SUMMARY FOR 2D, 9-POINT PATTERN
OPTIMAL VALUES OF PARAMETERS

	Free Run	Extrapolation	
Iteration Input Vector		No Forced Feedback MFB=0	Forced Feedback MFB=1
XOLD ONLY MON=0	Time = 3.393 NIT = 165	Min Time = 1.071 NIT = 36 ISK = 30 N = 5 Time Gain = 3.17 NIT Gain = 4.58	Min Time = 1.064 NIT = 36 ISK = 30 N = 4 Time Gain = 3.19 NIT Gain = 4.58
XOLD/XNEW (as available) MON=1	Time = 1.899 NIT = 90	Min Time = 0.790 NIT = 25 ISK = 20 N = 3 Time Gain = 2.41 NIT Gain = 3.60	(Not Done)

TABLE 3 - COMPUTATION TIMES AND ITERATION COUNTS 3D, 7-POINT PATTERN

TABLE 3A - MON=0, MFB=1

ISK \ N	1	2	3	4	5	6	7	8	9	10
0			52.340	44.368	40.207	38.889	38.763	36.694		37.737
			129	100	84	76	71	63		60
.										
.										
.										
40						20.163	17.370	17.506	18.201	
						67	61	59	60	
50			22.804			18.206	15.399	15.867	15.956	16.116
			86			70	64*	65	65	66
60						16.302	15.957	16.763	17.399	17.406
						73	72	73	74	74
.										
.										
.										
100			20.331				20.057			20.051
			109				107			107

TABLE 3 (Continued)

TABLE 3B - MON=1, MFB=0

ISK \ N	1	2	3	4	5	6	7	8	9	10
0			17.213	16.845	16.771	17.651				
	79	55	45	41	37	36				
.										
.										
.										
25					12.192	10.259				
					43	38				
30				9.856	8.391	8.448				
				41	37*	37				
35			9.605	8.660	8.643	8.635				
			44	41	41	41				
40			8.940	8.932						
			45	45						
45			9.832							
			50							

TABLE 3 (Continued)

TABLE 3C - SUMMARY FOR 3D, 7-POINT PATTERN
OPTIMAL VALUES OF PARAMETERS

	Free Run	Extrapolation	
Iteration Input Vector		No Forced Feedback MFB=0	Forced Feedback MBF=1
XOLD ONLY MON=0	Time = 50.092 NIT = 297	(Not Done)	Min Time = 15.399 NIT = 64 ISK = 50 N = 7 Time Gain = 3.25 NIT Gain = 4.64
XOLD/XNEW (as available) MON=1	Time = 23.621 NIT = 134	Min Time = 8.391 NIT = 37 ISK = 30 N = 5 Time Gain = 2.82 NIT Gain = 3.62	(Not Done)

MFB = 0 or 1: extrapolated vector, for which convergence criterion was not met, was not/was fed back as input into next iteration cycle

MON = 0 or 1: input to iteration is output of previous cycle/components of new output vector are fed in as generated.

For each combination of tactical options there is one table; for each problem there are up to four such tables plus a summary table showing the minimum computation time and corresponding iteration count as well as the gains achieved in comparison with free runs in which no extrapolation was attempted. (Time gain, for example, is the ratio of free run time to least time attained using extrapolation.)

In all cases the components of the starting vector were (the same sequence of) pseudorandom numbers in the range $-0.01 \leq x_i \leq +0.01$ and the criterion for convergence was based on $\varepsilon = TOL = 10^{-8}$.

For the problems considered, the optimum value of N lay in the range $3 \leq N \leq 7$, with N = 5 a good compromise. The procedure for determining this optimum N is to make a series of runs N = 1, 2, 3, ..., all with ISK = 0. That value of N for which computation time is either a minimum or nearly down to an asymptotic value is either the desired N or adjacent to it. The number of iterations corresponding to this N, for ISK = 0, is somewhat above the value of ISK which yields minimum time, but it gives a starting point from which to search for the optimal value. The best value of ISK is one that requires only one or two extrapolation cycles to obtain convergence, because the extra computation involved in extrapolation is relatively costly of time.

One interesting sidelight on tactical doctrine appears: when the optimal values of N and ISK are used, it makes relatively little difference whether MFB = 1 or 0, i.e., whether an extrapolated vector $\tilde{x}$, for which the convergence criterion has not been met, is or is not fed back as input to the next iteration.

By contrast, it stands out starkly that the very worst tactic is to force back rank one extrapolation (N=1) right from the start (ISK=0). This is easy to understand: the conditions under which rank one extrapolation is effective have not nearly been reached so that what actually takes place is merely a succession of restarts.

6. OTHER WORK ON EXTRAPOLATION OF VECTOR SEQUENCES

The best known and most influential work on the extrapolation of vector sequences is the vector epsilon algorithm of Peter Wynn (1962).[27] It was put forth as a generalization, by reinterpretation of his elegant scalar epsilon algorithm which he had proposed (Wynn (1956))[28] as a practical means of computing Shanks' transforms (Shanks (1955)).[7] Roughly speaking, if the zeroth column of the vector epsilon array contains the successive vectors from the iterative solution of a system of m linear equations, then each element of the $2m^{th}$ column is the solution vector. The exact theorem, taking into consideration the nature of the minimal polynomial of the iteration matrix, was discovered by Brezinski in 1972 (see Brezinski and Rieu (1974))[29] and by Gekeler (1972).[30] For a summary of results in this general area, see Brezinski's lecture notes (Brezinski (1977)).[31] His FORTRAN implementation of the vector epsilon algorithm, EPSV, plus discussion and worked examples, appears in his textbook on algorithms for accelerating convergence, Brezinski (1978).[32]

Unfortunately, the availability of this very useful product of mathematicians seems to be too little known to engineers. Acceleration of convergence could result in appreciable savings in the cost of their design calculations. Brezinski's text[32] is intended to help spread the word; so far it is available only in French.

Just as in the 1950's, when the needs of the nuclear energy industry provided impetus for developing the so-called relaxation methods for the iterative solution of large systems of linear equations, so in the 1970's the needs of the aerospace industry, particularly in solving problems in transonic flow, spurred several attempts to devise methods for accelerating convergence in the iterative solution of large linear systems.

In one series of papers, Martin and Lomax (1975),[33] and Martin (1975)[34] and (1976),[35] of NASA Ames Laboratory, the authors presented an ingenious method involving attaching to a power series the finite difference solutions of certain subsidiary problems arising from a perturbation analysis of the original partial differential equations. (The motivation was their understanding that Shanks had shown that a power series (even the first few terms!) tends to behave like a geometric series and hence can be summed by what they refer to as the "Aitken/ Shanks transform," (Equation (1.2) of this report. Thus they were led to

mesh-pointwise extrapolation of the first three perturbation functions as a means of accelerating convergence. For elaboration of these comments, see Eddy (1976).[36] Gains in computation time by a factor somewhat less than two are reported.

In another series of papers, Hafez and Cheng (1975),[37] (1976),[38] (1977)[39] of the University of Southern California at Los Angeles present the application to certain aerodynamic problems of their modification of Shanks' transforms (or equivalently, of Wynn's vector epsilon algorithm). Their first order transform handles the situation wherein the matrix of the linear approximation to the iteration equation has a single dominant eigenvalue, the second order transform copes with two dominant eigenvalues, and so on. They achieve a considerable reduction in required storage space by virtue of their theorem that, if the initial sequence of vectors is generated by a linear recursion, then so also is every even-ordered column of the corresponding vector epsilon array. They claim a reduction in computation time by a factor ranging from three to five.

Many papers have been written, by many authors, on the choice of the relaxation parameter, ω, in

$$\tilde{x} = (1-\omega)\ x_n + \omega\ x_{n+1} \tag{6.1}$$

As has already been pointed out in Section 4, rank one extrapolation of the present paper gives a particular way of calculating this parameter (Equations (4.26), (4.27)). Detailed comparisons of the relative effectiveness of this and other prescriptions have not been made and are not planned. Rank one extrapolation is relatively ineffective when compared to higher rank extrapolation, and much more interest attaches to possible relationships between rank r extrapolation for $r \geq 2$ and (a) the successive even-numbered columns of Wynn's vector epsilon array, or (b) the higher order transforms of Hafez and Cheng. These questions remain to be investigated. As has already been mentioned, something close to rank two extrapolation has been presented by Jennings (1971)[21] and by Hadjidimos (1978).[22]

After the manuscript of this report had been completed, (except for the remainder of this section), Claude Brezinski pointed out to the author that the ideas described herein are close to those contained in papers by Cabay and Jackson (1976),[40] Mešina (1977),[41] Smith and Ford (1980),[42] and Skelboe.[43] The main idea of the so-called polynomial method of Cabay and Jackson[40] can be sketched as follows, using the notation of the present paper:

Corresponding to the determination of the vector ξ (Equations (4.6), (4.7)) is the determination of a pseudo minimal polynomial

$$p(t) = \sum_{j=0}^{s+1} p_j \, t^g \qquad (p_{s+1}=1,\ p(1) \neq 0) \tag{6.2}$$

such that

$$||p(A)u_0|| = \left\| \sum_{i=0}^{s+1} p_i \, u_i \right\| \leq \varepsilon \tag{6.3}$$

Then the extrapolated vector is

$$\tilde{x} = x_0 + \sum_{i=0}^{s} c_i \, u_i \tag{6.4}$$

where

$$c_i = \left(\sum_{j=i+1}^{s+1} p_j \right) / p(1) \tag{6.5}$$

In spite of differences in approach and in mathematical machinery employed, the polynomial method and the reduced rank method are indeed quite parallel and share the same computational pitfalls.

Mešina's[41] approach is even closer to that of the present report. His extrapolated vector is again a linear combination of the iterated vectors

$$\tilde{x}_m = \sum_{j=0}^{m} c_j x_j \tag{6.6}$$

where the coefficients c_j are determined so as to minimize the ℓ_2 norm of

$$\tilde{u}_m = A\,\tilde{x}_m + b - \tilde{x}_m \tag{6.7}$$

$$= \sum_{j=0}^{m} c_j\,(u_j - b) + b \tag{6.8}$$

Here again the problem is to determine the rank, m, and the coefficients, c_j, which are subject to

$$\sum_{j=0}^{m} c_j = 1 \tag{6.9}$$

Mešina's convergence arguments lead to the same sort of polynomials that Cabay and Jackson employed. Further, he points out that his method is valid when a few eigenvalues of the iteration matrix, A, have absolute values greater than unity. He has applied his method to a problem in neutron transport theory, where the dimension of the system of equations is about 3000, and claims a reduction of iterative count by a factor ranging from 3 to 5. This claim is quite comparable to the claim of the present report.

Smith and Ford[42] combine the two foregoing methods into what they call the CJM method and apply it to a sequence of vectors generated by a nonlinear iteration in n space

$$x_{i+1} = F(x_i) \tag{6.10}$$

where F is Fréchet differentiable. Their extrapolation formula is

$$\tilde{x} = \sum_{i=0}^{k} c_i x_i \Big/ \sum_{i=0}^{k} c_i \quad (k \leq n) \tag{6.11}$$

where the c_i are the least squares solution of the overdetermined system

$$\sum_{i=0}^{k} c_i u_i = 0 \quad (c_k = 1) \tag{6.12}$$

and

$$u_i = x_{i+1} - x_i \tag{6.13}$$

They go on to show that the vector outputs of a sequence of iterative cycles, in which the output, $\tilde{x}$, of each cycle becomes the input, x_0, of the next cycle, behave exactly like the sequence of vectors produced in a similar fashion with the aid of Wynn's vector epsilon algorithm; hence, according to a result proved independently by Brezinski[29] and by Gekeler,[30] the vectors of this sequence converge quadratically in the ℓ_2 norm toward the solution of $x = F(x)$.

Their paper concludes with comparisons of the performance of the CJM method with the performance of the vector epsilon algorithm on a set of small test problems.

According to Smith (private communication), Skelbue's "algorithm is very similar to what we call CJM, and he proves a similar quadratic convergence result, albeit by a much longer proof."

It does indeed appear, as Brezinski pointed out, that the last few algorithms described above are essentially the same as the reduced rank algorithm presented herein. Apparently the time is ripe for such a step forward in this area of numerical analysis.

ACKNOWLEDGMENTS

My primary debt of gratitude is to Daniel Shanks who is credited by Gragg (1972)[44] with being partly responsible for initiating the surge of interest in extrapolation methods since 1949. It was my good fortune to have him as a colleague and friend at the start of my career when his discoveries were new and exciting; in addition to providing general intellectual stimulation, he acquainted me with his theory of extrapolation of sequences and with the related topics of continued fractions and the Padé table. Much later, when reading Martin and Lomax (1975)[33] had rearoused my interest in the extrapolation of vector sequences, he gave me free access to his magnificent library, called my attention to the extensive work of Peter Wynn on scalar and vector extrapolation which had sprung from his own work, and encouraged me to get in touch with Claude Brezinski who was becoming a champion of extrapolation methods.

To Claude Brezinski I am indebted for expressions of interest in my work and for encouragement, and also for copies of his reports, computer programs (including his implementation of Wynn's vector epsilon algorithm), and books.

T.M.E. Greville, H.K. Cheng, D.A. Smith, A.H. van Tuyl, and John Todd all sent me reports or reprints of their work in the theory or use of extrapolation techniques.

Ian Barrodale, who heard my oral presentation of the early version of reduced rank extrapolation (Eddy (1979)),[23] pointed out to me the usefulness of the ℓ_1 norm and sent me card decks for his ℓ_1 and ℓ_∞ optimization programs.

My colleague, Francis Henderson, supplied me with a card deck for the Businger-Golub singular value decomposition algorithm CSVD, which, modified to handle real arithmetic only, figured importantly in one phase of my computational experiments.

Mary Beth Marquardt and Richard van Eseltine helped me over the hurdles of learning to use a computer terminal (in place of both runs with decks of cards) to run my experimental computations.

APPENDIX

LISTING OF THE COMPUTER PROGRAMS

SIMP: Control routine

P3D: Subroutine for iterative solution of Laplace's equation

XTRP: Subroutine for reduced rank extrapolation

These programs are written in CDC FORTRAN Extended which is essentially FORTRAN IV.

```
C     LISTING FOR R P EDDY CODE 1843      8 APRIL 1988


      PROGRAM SIMP(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C     MFB = 0 TO IGNORE   UNSUCCESSFUL EXTRAPOLATION.
C     MFB = 1 TO FEEDBACK UNSUCCESSFUL EXTRAPOLATION.
C     MON = 0 FOR TOTAL  STEP ITERATION.  INPUT IS XOLD ONLY.
C     MON = 1 FOR SINGLE STEP ITERATION.  INPUT IS XOLD/XNEW AS AVAILABLE.
C     WITH N = 0, THIS DECK IS SET UP FOR A FREE RUN--NO EXTRAPOLATION.
      DIMENSION XOLD(1000), XNEW(1000), RS(1000)                        FOR P3D
      NR = 10
      NC = 10
      NL = 10                                                           FOR P3D
      M = NR*NC*NL                                                      FOR P3D
      NPPL = NR*NC                                                      FOR P3D
      TOL = 0.1E-7
      MXT = 500
      ISK = 0
      NSK =   0
      N   = 0
      MFB = 0
      MON = 0
   10 WRITE(6,1010)
 1010 FORMAT(58H1ACCELERATION OF CONVERGENCE BY REDUCED RANK EXTRAPOLATI
     1ON)
   11 WRITE(6,1011)
 1011 FORMAT(75H0LAPLACE EQUATION IN A PARALLELOPIPED WITH 10 ROWS 10 COFOR P3D
     1LUMNS AND 10 LAYERS)                                              FOR P3D
   12 WRITE(6,1012)
 1012 FORMAT(32H HOMOGENEOUS BOUNDARY CONDITIONS)
   13 WRITE(6,1013)
 1013 FORMAT(33H  7-POINT PATTERN   EQUAL WEIGHTS)                       7 POINT
   14 WRITE(6,1014)
 1014 FORMAT(80H COMPONENTS OF STARTING VECTOR ARE PSEUDO-RANDOM NUMBERS
     1 -0.01 .LE. X .LE. +0.01)
C     PARAMETER DO LOOP CONTROLS GO HERE.
   15 WRITE(6,1015) ISK,NSK,N,MFB,MON
 1015 FORMAT(11H0PARAMETERS,5X,5HISK =,I5,5X,5HNSK =,I3,7X,3HN =,I3,6X,5
     1HMFB =,I2,6X,5HMON =,I2)
C     GENERATE RIGHT SIDE VECTOR, RS
   20 DO 21 I=1,M
   21 RS(I) = 0.0
C     GENERATE STARTING VECTOR
C     PSEUDORANDOM NUMBERS IN THE RANGE  -.01 .LE. X .LE. +.01
   25 DO 26 I=1,M
   26 XOLD(I) = .02*RANF(I) - .01
C     BEGIN ORDINARY ITERATION CYCLE
      NIT = 1
      NEX = ISK
   30 CALL P3D(NR,NC,NL,NPPL,RS,XOLD,XNEW,MON)                          FOR P3D
C     CALCULATE L1, L2, AND LINF NORMS OF DIFFERENCE VECTOR
      R1 = 0.0
      R2 = 0.0
      R3 = 0.0
   35 DO 37 I=1,M
      W = XNEW(I) - XOLD(I)
      WA = ABS(W)
      R1 = R1 + WA
      R2 = R2 + W*W
```

```
   37 IF(WA .GT. R3) R3 = WA
      R2 = SQRT(R2)
C     TEST FOR CONVERGENCE
   40 IF(R2 - TOL) 41,41,50                                          L2  NORM
C     CONVERGENCE ATTAINED
   41 WRITE(6,1041) NIT
 1041 FORMAT(28H CONVERGENCE ATTAINED, NIT =,I5)
C
C     WRITE OUTPUT VECTOR, ETC, HERE.
C
      GO TO 99
   50 IF(NIT - NEX) 70,70,60
   60 CALL XTRP(M,N,XOLD,XNEW,TOL,NSK,NEX,NIT,MFB)
   70 DO 71 I=1,M
   71 XOLD(I) = XNEW(I)
   80 NIT = NIT + 1
      IF(NIT - MXT) 30,30,95
   95 WRITE(6,1095) NIT
 1095 FORMAT(6H0NIT =,I5,29H .GT. MXT WITHOUT CONVERGENCE)
   99 WRITE(6,1099)
 1099 FORMAT(9H0FINISHED)
      STOP
      END
```

```
C     LISTING FOR R P ECOY CODE 1843      8 APRIL 1988


      SUBROUTINE P3D(NR,NC,NL,NPPL,RS,XOLD,XNEW,MON)
C     7 POINT PATTERN
C     MON = 0 FOR TOTAL  STEP ITERATION.  INPUT IS XOLD ONLY.
C     MON = 1 FOR SINGLE STEP ITERATION.  INPUT IS XOLD/XNEW AS AVAILABLE.
      DIMENSION XOLD(1000), XNEW(1000), RS(1000)
      I = 0
      DO 32 KC=1,NL
      DO 31 JC=1,NC
      DO 30 IC=1,NR
      I = I+1
      W = RS(I)
      IF(MON) 10,15
   10 IF(IC.NE. 1) W=W+XNEW(I-1)
      IF(JC.NE. 1) W=W+XNEW(I-NR)
      IF(KC.NE. 1) W=W+XNEW(I-NPPL)
      GO TO 20
   15 IF(IC.NE. 1) W=W+XOLD(I-1)
      IF(JC.NE. 1) W=W+XOLD(I-NR)
      IF(KC.NE. 1) W=W+XOLD(I-NPPL)
   20 IF(IC.NE.NR) W=W+XOLD(I+1)
      IF(JC.NE.NC) W=W+XOLD(I+NR)
      IF(KC.NE.NL) W=W+XOLD(I+NPPL)
   30 XNEW(I) = W/6.
   31 CONTINUE
   32 CONTINUE
      RETURN
      END
```

```
C     LISTING FOR R P EDDY CODE 1843      8 APRIL 1980


      SUBROUTINE XTRP(M,N,XOLD,XNEW,TOL,NSK,NEX,NIT,MFB)
C     REDUCED RANK EXTRAPOLATION
C     THIS VERSION IS FOR USE WITH L2 AND SUCCESSIVE TRIANGULAR DECOMPOSITION
C     ALSO CALLED EXPANCING CHOLESKI.
C     MFB = 0 TO IGNORE   UNSUCCESSFUL EXTRAPOLATION.
C     MFB = 1 TO FEEDBACK UNSUCCESSFUL EXTRAPOLATION.
C
      DIMENSION XOLD(1000), XNEW(1000), XHLD(1000)                    FOR P30
      DIMENSION DIF1(1000,11), DIF2(1000,10)                          FOR P30
      DIMENSION XKSI(10), RSID(1000), HUTF(10,11)                     FOR P30
      DATA NCC,NCY /0,0/
C
C     BUILD UP FIRST AND SECOND ORDER DIFFERENCE MATRICES.
C
C     CALCULATE THE NCC+1 COLUMN OF DIF1
   10 DO 11 I=1,M
   11 DIF1(I,NCC+1) = XNEW(I) - XOLD(I)
      IF(NCC) 15,15,20
C     SAVE INITIAL VECTOR
   15 DO 16 I=1,M
   16 XHLD(I) = XOLD(I)
      NCC = NCC + 1
      GO TO 100
C     CALCULATE THE NCC COLUMN OF DIF2
   20 DO 21 I=1,M
   21 DIF2(I,NCC) = DIF1(I,NCC+1) - DIF1(I,NCC)
C
C     CALCULATE THE NCC COLUMN OF H*H
  200 DO 204 I=1,NCC
      W = 0.0
      DO 202 K=1,M
  202 W = W + DIF2(K,I)*DIF2(K,NCC)
  204 HUTF(I,NCC) = W
C     CALCULATE NCC ELEMENT OF H*U0
      W = 0.0
      DO 206 K=1,M
  206 W = W + DIF2(K,NCC)*DIF1(K,1)
      HUTF(NCC,N+1) = W
C     CALCULATE NCC COLUMN OF TRIANGULAR SQUARE ROOT
      IF(NCC - 1) 210,210,212
  210 HUTF(1,1) = SQRT(HUTF(1,1))
      HUTF(1,N+1) = HUTF(1,N+1)/HUTF(1,1)
      GO TO 230
  212 HUTF(1,NCC) = HUTF(1,NCC)/HUTF(1,1)
      DO 225 I=2,NCC
      IM = I - 1
      W = HUTF(I,NCC)
      DO 214 K=1,IM
  214 W = W - HUTF(K,I)*HUTF(K,NCC)
      IF(I - NCC) 216,218,218
  216 HUTF(I,NCC) = W/HUTF(I,I)
      GO TO 225
  218 IF(W) 220,220,224
  220 WRITE(6,1220) NCY,NCC,NIT,W
 1220 FORMAT(8HOTROUBLE,7X,5HNCY =,I3,5X,5HNCC =,I3,4X,5HNIT =,I4,6X,23H
     1DIAGONAL ELEMENT = SQRT,E13.6)
```

```
C     EXTRAPOLATE WITH LAST USABLE XKSI, THAT FOR NCC-1
      NCH = NCC - 1
C     SEE COMMENTS FOLLOWING LINE 72  BELOW.
      IF(MFB) 75,80
  224 HUTF(NCC,NCC) = SQRT(W)
  225 CONTINUE
C     CALCULATE NCC ELEMENT OF RIGHT SIDE
      W = HUTF(NCC,N+1)
      DO 226 K=1,IM
  226 W = W - HUTF(K,NCC)*HUTF(K,N+1)
      HUTF(NCC,N+1) = W/HUTF(NCC,NCC)
C     CALCULATE XKSI, THE VECTOR OF EXTRAPOLATION COEFFICIENTS
C     ROW L  HUTF(L,L)*XKSI(L) + SUM(HUTF(L,L+K)*XKSI(L+K) + HUTF(L,N+1) = 0
C                                1 .LE. K .LE. NCC-L = I
  230 XKSI(NCC) = -HUTF(NCC,N+1)/HUTF(NCC,NCC)
      DO 234 I=1,IM
      L = NCC - I
      W = -HUTF(L,N+1)
      DO 232 K=1,I
  232 W = W - HUTF(L,L+K)*XKSI(L+K)
  234 XKSI(L) = W/HUTF(L,L)
C
C     CALCULATE RSID, THE RESIDUAL VECTOR AND ITS NORMS UNDER L1,L2, LINF
   40 R1 = 0.0
      R2 = 0.0
      R3 = 0.0
   45 DO 48 I=1,M
      W = DIF1(I,1)
      DO 46 J=1,NCC
   46 W = W + DIF2(I,J)*XKSI(J)
      WA = ABS(W)
      R1 = R1 + WA
      R2 = R2 + W*W
      IF(WA .GT. R3) R3 = WA
   48 RSID(I) = W
      R2 = SQRT(R2)
C     TEST FOR CONVERGENCE
      NCH = NCC
   70 IF(R2 - TOL) 75,75,71                                          L2  TEST
C     ADVANCE COLUMN COUNTER, NCC
   71 NCC = NCC + 1
      IF(NCC - N) 100,100,72
   72 IF(MFB) 75,80
C     GO TO 75 TO FEED BACK UNSUCCESSFUL LAST EXTRAPOLATION
C     GO TO 80 TO IGNORE    UNSUCCESSFUL LAST EXTRAPOLATION
C     CALCULATE EXTRAPOLATED VECTOR XTLD AND PUT INTO XNEW
   75 DO 77 I=1,M
      W = XHLD(I)
      DO 76 J=1,NCH
   76 W = W + DIF1(I,J)*XKSI(J)
   77 XNEW(I) = W
C     RESET COLUMN COUNTER, NCC, FOR NEXT EXTRAPOLATION CYCLE
   80 NCC = 0
C     ADVANCE TEST COUNTER FOR NEXT EXTRAPOLATION CYCLE  (SIMP 50)
   85 NEX = NEX + NSK + N
  100 RETURN
      END
```

REFERENCES

1. Varga, R.S., "Matrix Iterative Analysis," Prentice-Hall, Englewood Cliffs, N.J. (1962).

2. Wachspress, E.L., "Iterative Solution of Elliptic Systems," Prentice Hall, Englewood Cliffs, N.J. (1966).

3. Young, D.M., "Iterative Solution of Large Linear Systems," Academic Press, New York (1971).

4. Forsythe, G.E. and C.B. Moler, "Computer Solutions of Linear Algebraic Systems," Prentice-Hall, Englewood Cliffs, N.J. (1967).

5. Aitken, A.C., "On Bernoulli's Numerical Solution of Algebraic Equations," Proceedings of the Royal Society of Edinburgh, Series A, Vol. 46, pp. 289-305 (1926).

6. Shanks, D., "An Analogy Between Transients and Mathematical Sequences and Some Nonlinear Sequence-to-Sequence Transforms Suggested by It. -- Part I," U.S. Naval Ordnance Laboratory Memorandum 9994 (26 July 1949).

7. Shanks, D., "Nonlinear Transformations of Divergent and Slowly Convergent Sequences," Journal of Mathematics and Physics (MIT), Vol. 34, No. 1, pp. 1-42 (1955).

8. Barrodale, I. and F.D.K. Roberts, "Algorithm 478: Solution of an Overdetermined System of Equations in the ℓ_1 Norm," Communications of the Association for Computing Machinery, Vol. 17, pp. 319-320 (1974).

9. Barrodale, I. and C. Phillips, "Algorithm 495: Solution of an Overdetermined System of Linear Equations in the Chebyshev Norm," ACM Transactions on Mathematical Software, Vol. 1, No. 3, pp. 264-270 (1975).

10. Barrodale, I. and F.D.K. Roberts, "An Efficient Algorithm for Discrete ℓ_1 Linear Approximation With Linear Constraints," SIAM Journal on Numerical Analysis, Vol. 15, No. 3, pp. 603-611 (1978).

11. Bartels, R.H. and G.H. Golub, "Stable Numerical Methods for Obtaining the Chebyshev Solution to an Overdetermined System of Equations," Communications of the ACM, Vol. 11, No. 6, pp. 403-408 (1968).

12. Bartels, R.H. and G.H. Golub, "Algorithm 328: Chebyshev Solution to an Overdetermined System," Communications of the ACM, Vol. 11, No. 6, pp. 428-430 (1968).

13. Bartels, R.H. et al., "On Cline's Method for Solving Overdetermined Linear Systems in the ℓ_∞ Sense," SIAM Journal on Numerical Analysis, Vol. 15, No. 2, pp. 255-270 (1978).

14. Bartels, R.H. et al., "Minimization Techniques for Piecewise Differentiable Functions: The ℓ_1 Solution to an Overdetermined Linear System," SIAM Journal on Numerical Analysis, Vol. 15, No. 2, pp. 224-241 (1978).

15. Peters, G. and J.H. Wilkinson, "The Least Squares Problem and Pseudo-Inverses," The Computer Journal, Vol. 13, pp. 309-316 (1970).

16. Ben-Israel, A. and T.N.E. Greville, "Generalized Inverses: Theory and Applications," John Wiley and Sons, New York (1974).

17. Boullion, T.L. and P.L. Odell, "Generalized Inverse Matrices," Wiley-Interscience, New York (1971).

18. Lawson, C.L. and R.J. Hanson, "Solving Least Squares Problems," Prentice-Hall, Englewood Cliffs, N.J. (1974).

19. Greville, T.N.E., "Some Applications of the Pseudo-Inverse of a Matrix," SIAM Review, Vol. 2, No. 1, pp. 15-22 (1960).

20. Greville, T.N.E., "Note on the Generalized Inverse of a Matrix Product," SIAM Review, Vol. 8, No. 4, pp. 518-521 (1966).

21. Jennings, A., "Accelerating the Convergence of Matrix Iterative Processes," Journal of the Institute of Mathematics and Its Applications, Vol. 8, No. 1, pp. 99-110 (1971).

22. Hadjidimos, A., "Accelerated Overrelaxation Method," Mathematics of Computation, Vol. 32, No. 141, pp. 149-157 (1978).

23. Eddy, R.P., "Extrapolating to the Limit of a Vector Sequence," pp. 387-396 of Wang, P.C.C. et al. (editors), "Information Linkage Between Applied Mathematics and Industry," (Proceedings of a Workshop, Naval Postgraduate School, Monterey, CA (Feb 1978), Academic Press, New York (1979).

24. Stewart, G.W., "Introduction to Matrix Computations," Academic Press, New York (1973).

25. Golub, G.H. and C. Reinsch, "Singular Value Decomposition and Least Squares Solutions," Numerische Mathematik, Vol. 14, No. 5, pp. 403-420 (1970). Also appears on pp. 134-151 of Wilkinson, J.H. and C. Reinsch (Editors), "Linear Algebra," Handbook for Automatic Computation, Vol. 2, Grundlehren der Mathematischen Wissenschaften, Band 186, Springer-Verlag, Berlin-Heidelberg-New York (1971).

26. Businger, P.A. and G.H. Golub, "Algorithm 358: Singular Value Decomposition of a Complex Matrix," Communications of the ACM, Vol. 12, No. 10, pp. 564-565 (1969).

27. Wynn, P.,"Acceleration Techniques for Iterated Vector and Matrix Problems," Mathematics of Computation, Vol. 16, No. 79, pp. 301-322 (1962).

28. Wynn, P., "On a Device for Computing the $e_m(S_n)$ Transform," Mathematical Tables and Other Aids to Computation (= Mathematics of Computation), Vol. 10, No. 54, pp. 91-96 (1956).

29. Brezinski, C. and A.C. Rieu, "The Solution of Systems of Equations Using the ε-Algorithm, an Application to Boundary-Value Problems," Mathematics of Computation, Vol 28, No. 127, pp. 731-741 (1974).

30. Gekeler, E., "On the Solution of Systems of Equations by the Epsilon Algorithm of Wynn," Mathematics of Computation, Vol 26, No. 118, pp. 427-436 (1972).

31. Brezinski, C., "Accélération de la Convergence en Analyse Numérique," Lecture Notes in Mathematics 584, Springer-Verlag, Berlin-Heidelberg-New York (1977).

32. Brezinski, C., "Algorithmes d'Accélération de la Convergence E´tude Numérique," Editions Technip, Paris (1978).

33. Martin, E.D. and H. Lomax, "Rapid Finite-Difference Computation of Subsonic and Slightly Supercritical Aerodynamic Flows," American Institute of Aeronautics and Astronautics Journal, Vol. 13, No. 5, pp. 579-586 (1975).

34. Martin, E.D., "Progress in Application of Direct Elliptic Solvers to Transonic Flow Computations," pp. 839-887 of Aerodynamic Analyses Requiring Advanced Computers--Part II, NASA SP-347 (1975).

35. Martin, E.D., "A Technique for Accelerating Iterative Convergence in Numerical Integration, With Application in Transonic Aerodynamics," pp. 123-139 of Cabannes, H. (Editor), "Padé Approximants Method and Its Applications to Mechanics" (Colloque Euromech 58, Toulon, France, 1975), Lecture Notes in Physics 47, Springer-Verlag, Berlin-Heidelberg-New York (1976).

36. Eddy, R.P., "Comment on 'Rapid Finite-Difference Computation of Subsonic and Slightly Supercritical Aerodynamic Flows'," American Institute of Aeronautics and Astronautics Journal, Vol. 14, No. 8, pp. 1149-1151 (1976).

37. Hafez, M.M. and H.K. Cheng, "Convergence Acceleration and Shock Fitting for Transonic Aerodynamic Computations," University of Southern California, Department of Aerospace Engineering, Report USCAE-132 (Apr 1975).

38. Cheng, H.K. and M.M. Hafez, "Cyclic Iterative Method Applied to Transonic Flow Analysis," pp. 101-121 of Cabannes, H. (Editor) (1976). See Reference 35 above.

39. Hafez, M.M. and H.K. Cheng, "Convergence Acceleration of Relaxation Solutions for Transonic Flow Computations," American Institute of Aeronautics and Astronautics Journal, Vol. 15, No. 3, pp. 329-336 (1977).

40. Cabay, S. and L.W. Jackson, "A Polynomial Extrapolation Method for Finding Limits and Antilimits of Vector Sequences," SIAM Journal on Numerical Analysis,, Vol. 13, No. 5, pp. 734-752 (1976).

41. Mešina, M., "Convergence Acceleration for the Iterative Solution of the Equation x = Ax + f," Computer Methods in Applied Mechanics and Engineering, Vol. 10, No. 1, pp. 165-173 (1977).

42. Smith, D.A. and W.F. Ford, "On the Cabay-Jackson-Mesina Algorithm," (Paper 773-65-3 presented at the meeting of the American Mathematical Society, San Antonio, Texas) (3-8 Jan 1980).

43. Skelboe, S., "Computation of the Steady State Response of Nonlinear Networks by Extrapolation Methods," IEEE Transactions on Circuits and Systems, Vol. 27, pp. 161-175 (Mar 1980).

44. Gragg, W.B., "The Padé Table and Its Relation to Certain Algorithms of Numerical Analysis," SIAM Review, Vol. 14, No. 1, pp. 1-62 (1972).

INITIAL DISTRIBUTION

Copies

4 CNR
- 1 ONR 430
- 1 ONR 430C/M. Cooper
- 1 ONR 432/S. Brodsky
- 1 ONR 438/R.D. Cooper

1 NRL/Lib

1 USNA/Lib

3 NAVSEA
- 1 SEA 03F/J.L. Schuler
- 1 SEA 321/R.G. Keane
- 1 SEA 996/Technical Lib

1 NAVPGSCOL/Lib

2 NSWC/MD
- 1 Dr. A.H. van Tuyl
- 1 Library

1 NSWC/VA/Lib

12 DTIC

2 NASA/AMES
- 1 Dr. Harvard Lomax
- 1 Library

1 NASA/Lewis Res Cen
Dr. William F. Ford

1 DOE/ARGONNE/Lib

1 BRIGHAM YOUNG UNIV
MAYO CLINIC
Attn: Dr. Arnold Lent
Rochester, MN 55901

1 BROWN UNIV/APPLIED MATH DIV
Prof. Philip J. Davis

2 CALIFORNIA INST OF TECH/
DEPT OF MATH
- 1 Prof. Herbert B. Keller
- 1 Prof. John Todd

Copies

1 UNIV OF SOUTHERN CALIFORNIA
DEPT OF AEROSPACE ENG/Prof. H.K.
Cheng

1 DUKE UNIVERSITY/DEPT OF MATH
Prof. David Smith

1 UNIV OF KENTUCKY/DEPT OF MATH
Prof. William B. Gragg, Jr.

1 UNIV OF MARYLAND/COMPUTER SCI DEPT
Prof. G.W. Stewart

1 UNIV OF MARYLAND/DEPT OF MATH
Dr. Daniel Shanks

1 NORTHWESTERN UNIV/COMPUTER SCI
DEPT/Prof. Erwin H. Barerss

1 UNIV OF WISCONSIN/DEPT OF
STATISTICS/Prof. Grace Wagba

1 COMPUTER SCIENCE DEPT
SUNY CENTER AT BUFFALO
Dr. Yair Censor
4226 Ridge Lea Road
Amherst, NY 14226

1 GENERAL ELECTRIC COMPANY
KNOLLS ATOMIC POWER LAB
Dr. Eugene L. Wachspress
PO Box 1072
Schenectady, NY 12301

1 Prof. Gene H. Golub
SERRA HOUSE, Serra Street
Stanford, CA 94305

1 Prof. T.N.E. Greville
700 Glenview Drive
Madison, WI 53716

1 LOS ALAMOS SCI LAB/Lib

1 Dr. Harry Polachek
11801 Rockville Pike
Rockville, MD 20852

Copies

1 Dr. Patrick J. Roache
4925 Kathryn Circle SE
Albuquerque, NM 87108

1 SANDIA Lib/LIVERMORE/Lib

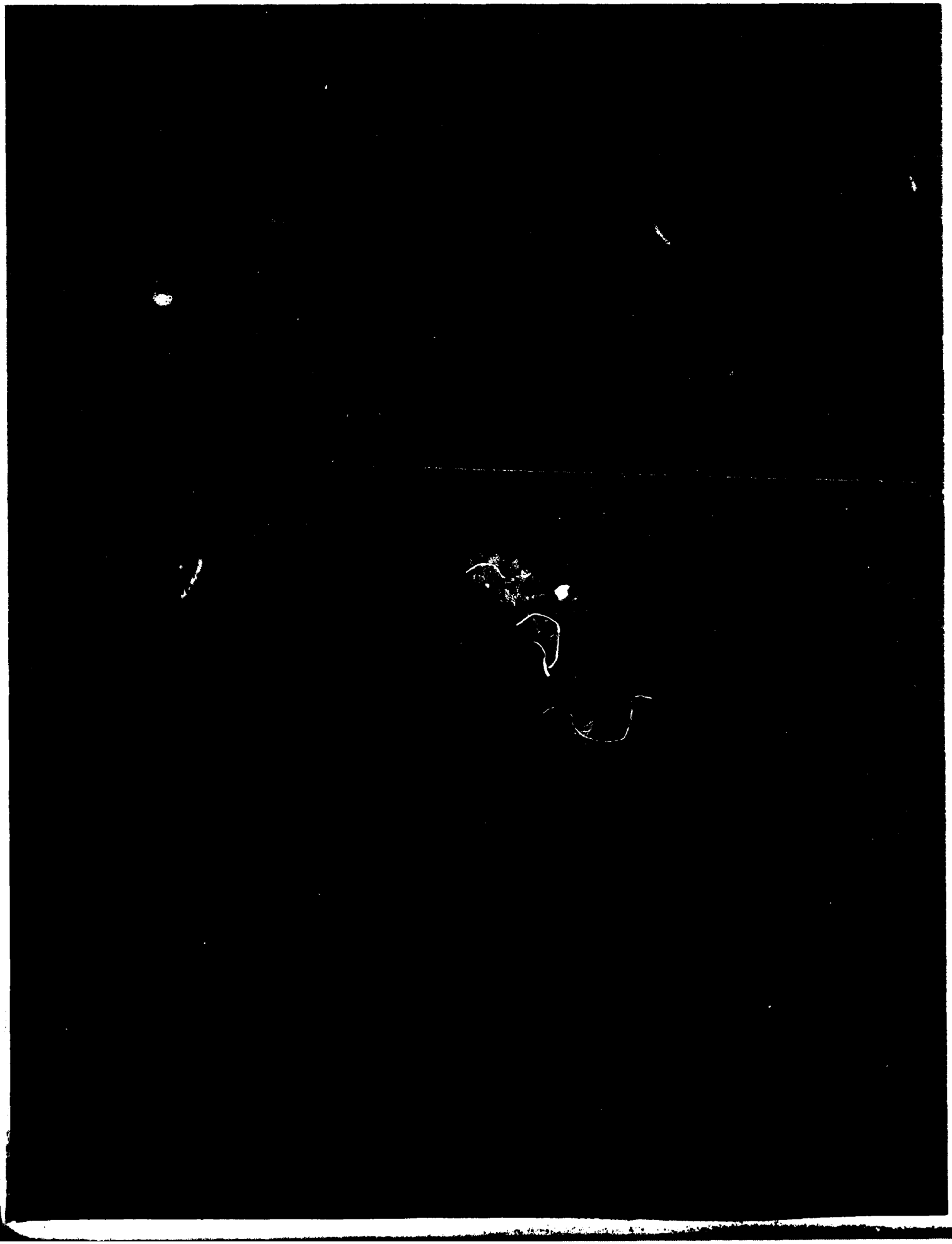
1 Dr. Feudor Theilheimer
2608 Spencer Road
Chevy Chase, MD 20015

1 Prof. Richard S. Varga
7065 Arcadia Drive
Parma, OH 44129

1 Mr. Thomas S. Walton
9816 Rosensteel Ave
Silver Spring, MD 20910

CENTER DISTRIBUTION

Copies	Code	Name
1	1552	J.H. McCarthy
1	16	Dr. H.R. Chaplin
1	1805	Dr. E.H. Cuthill
2	1809.3	D.K. Harris
1	1826	L.K. Meals
1	184	J.W. Schot
1	184.1	H. Feingold
2	1843	Dr. H.J. Haussling
1	1843	R.T. van Eseltine
10	1843	R.P. Eddy
1	1844	Dr. G.C. Everstine
1	1844	D.A. Gignac
1	1844	F.M. Henderson
1	1844	E.A. Schroeder
1	1870	P.L. Price
10	5211.1	Reports Distribution
1	522.1	Unclassified Lib (C)
1	522.2	Unclassified Lib (A)




FILMED
2-8